

UNIVERSITATEA POLITEHNICA BUCUREȘTI  
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE  
DEPARTAMENTUL CALCULATOARE



## PROIECT DE DIPLOMĂ

### **Aplicație web pentru managementul învățării** versiunea 2024

Panait Ștefan-Andrei

**Coordonator științific:**  
Prof. dr. ing. Andrei Ionescu

**BUCUREȘTI**

2024

UNIVERSITY POLITEHNICA OF BUCHAREST  
FACULTY OF AUTOMATIC CONTROL AND COMPUTERS  
COMPUTER SCIENCE DEPARTMENT



## DIPLOMA PROJECT

Aplicație web pentru managementul învățării  
2024 version

Ioana Popescu

**Thesis advisor:**  
Prof. dr. ing. Andrei Ionescu

**BUCHAREST**

2024

## CUPRINS

Sinopsis .....	2
Abstract.....	<b>Error! Bookmark not defined.</b>
Mulțumiri .....	3
1    Introducere .....	4
1.1    Context.....	4
1.2    Problema.....	4
1.3    Obiective .....	5
1.4    Structura lucrării .....	4
2    Analiza și specificarea cerințelor.....	7
3    Studiu de piață / Abordări existente.....	11
3.1    Indicații formatare figuri.....	<b>Error! Bookmark not defined.</b>
4    Soluția propusă .....	17
4.1    Indicații formatare formule .....	<b>Error! Bookmark not defined.</b>
5    Detalii de implementare .....	27
5.1    Indicații formatare tabele .....	27
6    Studiu de caz / Evaluarea rezultatelor .....	29
7    Concluzii .....	31
8    Bibliografie .....	32
9    Anexe .....	34

## **SINOPSIS**

Aplicația web pentru managementul învățării este o platformă ușor de utilizat pentru a facilita educației și a studiul suplimentar, online. Această are o interfață prietenoasă fiind foarte fiabilă atât pentru profesori cât și pentru cursanți în vederea interacționării, gestionării cursurilor. Cursanții înscriși pot să primească recenzii personalizate și să acceseze diferite resurse materiale educative.

Aplicația oferă siguranța utilizatorilor prin autentificare securizată și controlul accesului pe baza de rol pentru modificarea permisiunilor la funcționalități. Printre alte opțiuni, utilizatorul își poate personaliza profilul, atât în rolul de student cât și de instructor. Profesorii pot edita modulele cursurilor și emblema, pot da teme suplimentare și materiale elevilor, precum și recenzii personalizate și nota după verificarea temelor efectuate de cursanți.

## **MULȚUMIRI**

(opțional) Aici puteți introduce o secțiunea specială de mulțumiri / acknowledgments.

## **1 INTRODUCERE**

În domeniul educației, nevoia de inovație este constantă. Atât contextul mondial cât și tehnologia dezvoltată din ultimii ani au atins inevitabil și anii de temelie din viața fiecărui individ, mai exact, anii de școală astfel, modernizarea învățământului reprezintă o prioritate. Cu o tranziție neprevăzută și foarte rapidă în domeniul învățământului, profesorii și elevii au fost și sunt nevoiți să utilizeze diferite platforme online pentru continuarea predării și a studiului. Este evident faptul că nevoia de performanță, flexibilitate, cooperare și accesibilitate nu a fost niciodată la fel de mare.

Aplicația de management al învățării pe care o prezint se adresează acestor nevoi, oferind un mediu online complex pentru facilitarea și sprijinul educației și al dezvoltării. Această platformă este creată pentru îmbunătățirea experienței educaționale a studenților, a profesorilor, dar și a unităților de învățământ.

La bază, aplicația oferă o interfață centralizată, prietenoasă cu utilizatorii și clară, care favorizează comunicarea dintre profesori și cursanți. Cu capacități ce variază de la crearea cursului, gestionarea și editarea acestuia până la livrarea de conținut educațional suplimentar și evaluare detaliată și notarea temelor efectuate și trimise profesorului prin intermediul platformei, aplicația este de un real ajutor în interacțiunea profesor-cursant și facilitarea înțelegerii noțiunilor precum și suplimentarea orelor standard. Focalizându-se pe accesibilitate și performanță la nivel înalt, aplicația web asigură utilizatorilor navigarea cu ușurință pe platformă, accesarea resurselor materiale. Astfel, aplicația asigură elevilor siguranța că educația lor nu este afectată și pot continua călătoria lor de dezvoltare constant, indiferent de factorii care pot părea că împiedică continuitatea educației.

De altfel, consider că fiabilitatea și ușurința folosirii aplicației pot fi cele mai importante atuuri deoarece astfel poate fi la îndemână oricui, fără a fi nevoie de timp suplimentar special pentru înțelegerea și învățarea folosirii aplicației.

### **1.1 Context**

Situațiile neprevăzute din ultimii ani au dus către o schimbare drastică a organizării învățământului, de la metodele tradiționale de educație, la un nou tip de educație, cel online, care încă tinde să se dezvolte din ce în ce mai mult. Această schimbare a evidențiat importanța și nevoia unei platforme digitale bine dezvoltate ușor de folosit în vederea asigurării continuității educației.

### **1.2 Problema**

Învățarea suplimentară poate deveni greu de realizat în contextul tradițional în care profesorii și elevii sunt constrânși și nu își pot desfășura orele și materialele după cum doresc adaptându-se în funcție de fiecare elev în parte. În continuare voi enumera o parte din inconveniente care restricționează învățarea suplimentară în contextul tradițional:

- Limitările de timp
- Limitările de spațiu
- Numărul redus al resurselor materiale puse la dispoziție

- Lipsa interactivitatii si modernizarii
- Lipsa recenziilor detaliate si personalizate
- Lipsa flexibilitatii
- Lipsa temelor personalizate in functie de nevoile elevilor
- Constrangerea folosirii cartilor impuse de scoala

### 1.3 Obiective

Obiectivul principal pe care mi l-am propus in crearea unei noi platforme de invatare, in scopul invatarii suplimentare, il constituie facilitarea, sprijinirea, inovarea si imbunatatirea calitatii invatarii suplimentare, atat pentru studenti cat si pentru profesori.

Consider ca invatarea suplimentara trebuie sa fie la indemana oricui si usor de accesat si folosit, de aceea vdoresc sa evidentialez o parte dintre beneficiile pe care platforma le ofera profesorilor si studentilor.

#### a) Beneficii pentru profesor:

1. Flexibilitate in adaugarea resurselor materiale educative: Profesorii pot incarca oricand materialele educative care le sunt disponibile cursantilor in orice moment. De asemenea profesorii pot adauga materiale selectate de ei, nu doar din cartile de la scoala.
2. Crearea si gestionarea temelor: Profesorii pot adauga diferite probleme in lista de exercitii, de asemenea acestia pot select orice problema din lista pentru a crea o tema pentru elevi. Temele pot fi personalizate pentru nevoile comune ale mai multor studenti sau personalizate pentru un anumit elev. Profesorul poate impune sau nu un anumit termen limita pentru realizarea si incarcarea temelor de catre elevi.
3. Evaluarea temelor: Dupa efectuarea si incarcarea temelor de catre elevi, profesorii le pot corecta oricand. Corectarea temelor se poate realizeaza prin citirea lor si oferirea de recenzii personalizate fiecarui elev, cu exemple specifice din tema sa, incluzand si nota.

#### b) Beneficii pentru studenti:

1. Flexibilitatea in accesarea materialelor educationale oferite de profesori: Studentii pot vizualiza si descarca resursele materiale oferite de profesori si disponibile in orice moment.
2. Incarcarea temelor: Elevii pot incarca temele efectuate asigurandu-se ca lucrarile lor urmeaza sa fie evaluate de profesori si sa primeasca comentarii clare si precise cu greselile facute si sugestiile primite de catre cadrele didactice. De asemenea elevii dispun de flexibilitatea si in efectuarea si incarcarea pe platforma a temelor rezolvate.
3. Evaluare: Dupa evaluare temelor, elevii pot citi notitele si sugestiile profesorilor in timp ce isi pot revedea lucrarea, cat si rezolvarea corecta a acesteia. De asemenea fiecare elev primeste nota si recenzia in mod privat.

### 1.4 Structura lucrării

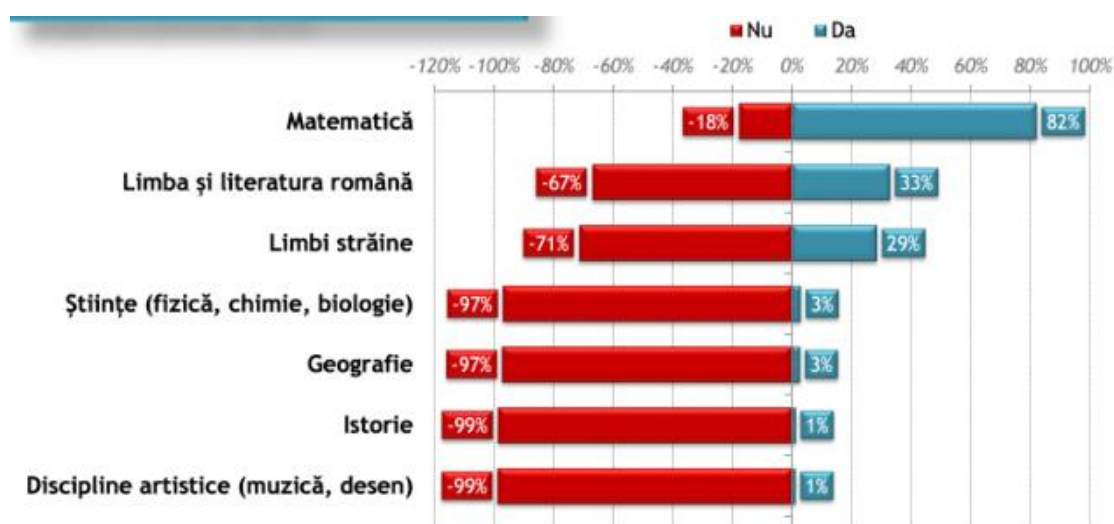
Un paragraf în care fiecare dintre secțiunile următoare este prezentată în 1-2 fraze, punând accentul pe elementele cele mai semnificative din fiecare secțiune.





## 2 ANALIZA ȘI SPECIFICAREA CERINȚELOR

Putem remarca faptul ca in Romania cultura meditatiilor este una foarte dezvoltata, fiecare parinte isi trimite copilul la meditatie macar in cei mai importanti ani din ciclul de invatamant al copiilor acestora. Un studiu la nivel national realizat de IRES si Societatea Academica din Romania spune ca „Conform studiului realizat de IRES și SAR în rândul părinților, o treime dintre elevi participă la sistemul de meditații, cu o frecvență considerabilă în clasele terminale, atunci când examenele naționale se apropie. În urma chestionării elevilor de gimnaziu și liceu, a rezultat faptul că un sfert dintre elevii din România accesează sistemul de meditații la cel puțin o materie.”<sup>1</sup>



Comparand aceste informatii cu alte tari:

- In Italia 1 din 5 elevi fac meditatie conform unei statistici realizate pe 6000 de elevi de repetizioni.it, platforma italiana pentru mediatii atat online cat si fizic.<sup>2</sup>
- In germania la fel ca in italia 1 din 5 elevi se adreseaza profesorilor pentru meditatie in invatamantul liceal conform lucrarii stiintifice „Effectiveness of private tutoring during secondary schooling in Germany: Do the duration of private tutoring and tutor qualification affect school achievement?”<sup>3</sup>

Din aceste statistici putem concluziona ca studentilor nu le sunt de ajuns orele de pregatire din cadrul scolilor. Astfel aceasta platforma ar actiona ca un pilon educational pentru student.

<sup>1</sup> <https://www.edupedu.ro/piata-meditatiilor-estimata-la-peste-120-de-milioane-de-lei-pe-luna-pe-baza-datelor-publice-si-a-cercetarilor-recente-analiza/>

<sup>2</sup> <https://www.rainews.it/articoli/2024/04/scuola-1-studente-su-5-va-a-ripetizioni-cosi-le-famiglie-spendono-circa-450-euro-7a461864-a445-4a71-849c-ff05fbf738b6.html#:~:text=Il%20trend%20C3%A8%20confortante%2C%20sia,si%20C3%A8%20passati%20al%202024%25>

<sup>3</sup> <https://www.sciencedirect.com/science/article/pii/S0959475219303093>

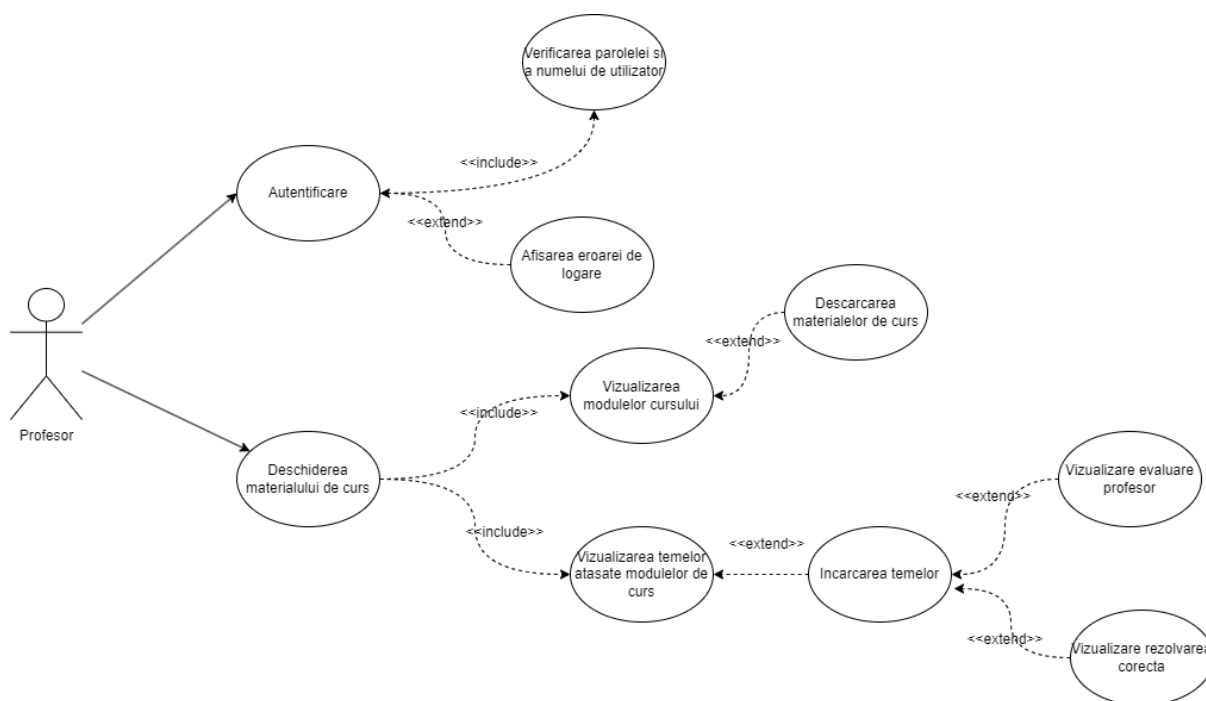
## 2.1 Cerinte nonfunctionale

Cerintele nonfunctionale se refera la specificatiile care descriu cum ar trebui sa se comporte un sistem, acestea se concentreaza pe calitatea si performanta aplicatiei si sunt esentiale pentru a asigura o experienta optima pentru utilizatori.

1. Securitate: datele utilizatorilor sunt protejate la orice moment de timp
2. Performanta: timpii de raspuns si viteza aplicatiei sa nu fie un impediment in folosirea acesteia
3. Viteza: Aplicatia trebuie sa raspunda cerintelor intr-un timp optim.
4. Fiabilitate: Aplicatia v-a functiona fara erori care sa afecteze disponibilitatea aplicatiei.

## 2.2 Cerinte functionale

Cerintele functionale se axeaza pe cum aplicatia functioneaza daca atinge sau nu anumite caracteristici dorite. In aplicatia propusa sunt 3 roluri: student, profesor si administrator. In continuare voi prezenta procesul de folosire a aplicatiei atasand diagrame ale cazurilor de utilizare.



In rolul de student procesul este urmatorul:

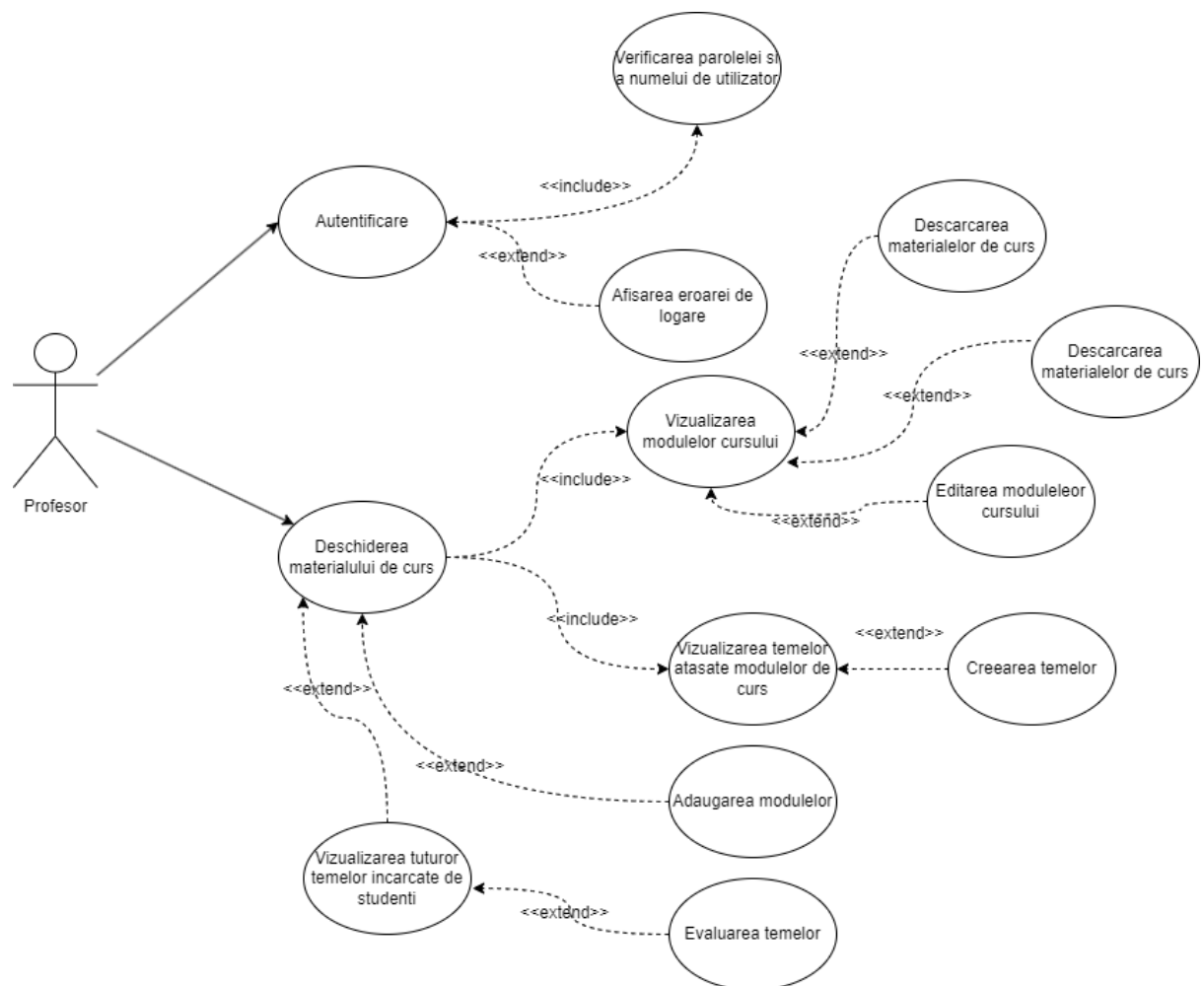
1. Crearea contului: Oricarei creare de cont nou ii este atribuit automat rolul de student.
2. Autentificarea: La autentificare studentul isi adauga credentialele urmand ca acestea sa fie verificate. Daca acestea sunt corecte utilizatorul este redirectat pe pagina unde sunt vizibile cursurile. Daca credentialele sunt incorecte acestuia ii va fi afisat un mesaj de eroare.

3. Inrolare la cursuri: Utilizatorul se poate inrola singur la cursurile la care crede ca are nevoie de studiu suplimentar.
4. Deschiderea materialului de curs: Dupa inrolarea la curs este accesibila deschiderea si privirea materialelor disponibile.
5. Vizualizarea modulelor cursului: Dupa deschiderea materialului de curs sunt vizibile modulele cursului. Un exemplu de modul: Titlul: „Polinoame”, Descriere: „Acest modul te va ajuta sa intelegi polinoamele” si un fisier atasat Suport curs: polinoame.pdf.
6. Descarcarea suportului de curs: Studentul poate descarca suportul de curs.
7. Vizualizarea temelor atasate modulelor de curs: Pentru fiecare modul cursantul poate avea atasat o tema.
8. Incarcarea temelor: Dupa ce studentul considera ca tema este rezolvata acesta poate incarca tema pentru evaluarea unui profesor.
9. Vizualizare evaluare profesor: Elevul poate observa comentariile profesorului asupra temei incarcate.
10. Vizualizare rezolvare corecta: Dupa evaluarea temei de catre profesor rezolvarea corecta a exercitiului este disponibila elevului.



In rolul de administrator procesul este urmatorul:

1. Creerea de utilizator: administratorul poate crea utilizatori
2. Modificarea informatiilor utilizatorilor cat si a drepturilor de student sau profesor: administratorul poate schimba informatiile utilizatorului de ex numele, rolul sau adresa de email.
3. Stergerea de utilizatori
4. Creerea de cursuri: administratorul poate crea cursuri
5. Modificarea informatiilor cursurilor: Administratorul poate schimba titlul, descrierea cat si categoria din care face parte cursul
6. Stergerea de cursuri
7. Inrolarea de studenti sau profesori la cursuri: Elevul se poate inrola si singur la cursuri dar profesorul se poate inrola doar daca este inrolat de administrator. Ratiunea din spatele acestei functionalitati este ca doar un administrator poate stii daca un profesor este bun la o materie. Astfel nu se v-a intampla de exemplu ca un profesor de Informatica sa corecteze si sa puna materiale la Geografie.
8. Vizualizare lista de probleme: Administratorul poate vizualiza lista de probleme, acesta poate edita problemele sau de asemenea sterge probleme.
9. Adaugare de probleme: Administratorul de asemenea poate adauga probleme.



În rolul de profesor procesul este următorul:

1. Editarea siglei de curs: profesorul poate schimba sigla la cursurile la care acesta este înrolat.
2. Adăugarea de probleme în lista de probleme: profesorul poate adăuga probleme
3. Adăugarea modulelor cursurilor: acesta poate adăuga module noi pentru curs
4. Editarea modulelor de curs: Acesta poate edita titlul modulului, descrierea, cât și poate șterge sau încărca suportul de curs aferent modulului.
5. Vizualizare lista de probleme: Profesorul poate vizualiza lista de probleme pentru a alege o problemă din aceeași listă pe care să o dea ca temă.
6. Adăugare de probleme: Acesta poate să adăuge probleme.
7. Creare de temă: Profesorul poate crea teme alegând o problemă din lista de probleme. Acesta poate de asemenea să seteze un număr de zile limită pentru încărcarea acestora.
8. Vizualizarea listei cu elevii care au încărcat temele: Profesorul poate vizualiza lista cu cei care au încărcat temele, apoi cu ajutorul unui modul de căutare acesta poate căuta după mai multe criterii cum ar fi titlul temei sau numele elevului.
9. Evaluarea temei: Acesta poate da o recenzie temelor încărcate cât și o notă.

### **3 STUDIUL DE PIAȚĂ / ABORDĂRI EXISTENTE**

Obiectivul apariției unui produs nou pe piață trebuie să se rezume în totalitate la rezolvarea problemelor potențialilor utilizatori, astfel încât fiecare aplicație disponibilă este orientată unui anumit grup țintă.

Aplicația pe care o propun se orientează către elevi și profesori, cu scopul facilitării procesului de învățare și de predare. Consider că, la momentul actual, în special în România, ideea de meditații este puternic răspândită printre elevi, de mulți ori numărul cererilor depășind numărul ofertelor.

De altfel, din cauza contextelor globale din ultimii ani, profesorii au fost nevoiți să se orienteze către metode noi de învățare, mai exact platformele de învățare online, considerate poate chiar „nonconformiste” până în acel moment. Pe parcurs, obișnuindu-se cu noile metode, din ce în ce mai mulți profesori au început să utilizeze platformele online de învățare, considerându-le benefice.

În domeniul educației este nevoie tot timpul de metode noi și inovative care să faciliteze învățarea. Astfel, pentru a realiza o platformă online de învățare suplimentară, utilă și fiabilă, am creat un chestionar orientat către nevoile întâmpinate de elevi, aflând de asemenea și părerea unui număr de 65 de persoane despre ideea de „platformă online de învățare”.

Ce vârstă aveți?

65 responses

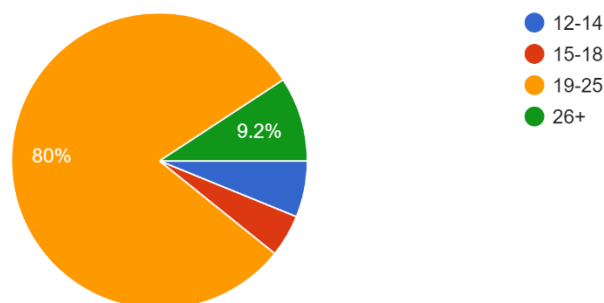


Figura 1

Mediul de proveniență

65 responses

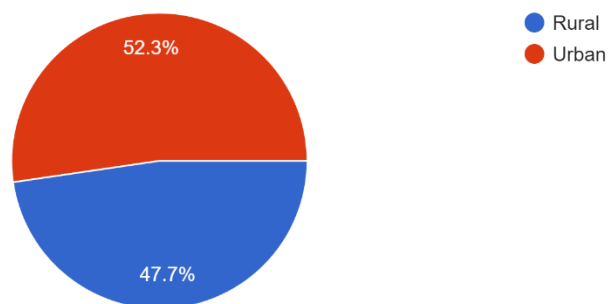


Figura 2

Cu ajutorul figurilor 1 și 2 putem observa fiabilitatea platformelor online de învățare, relevante și benefice persoanelor de diferite vârste, fiind chestionate 65 de persoane cu vârste cuprinse între 12 și peste 26 de ani, provenite atât din mediul urban (52.3%) cât și din cel rural (47.7%)

Ultima formă de învățământ finalizată  
65 responses

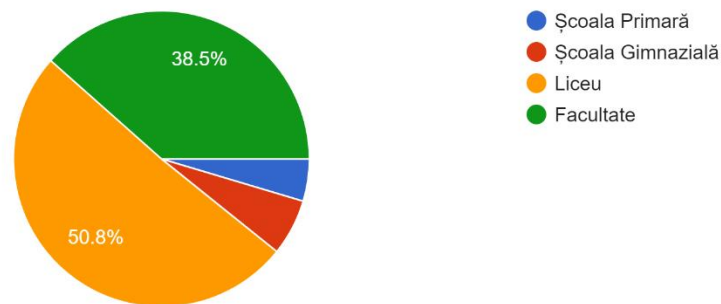


Figura 3

În perioada în care ați fost student/elev, ați simțit nevoia de ore suplimentare, în plus față de cele de la școală?  
65 responses

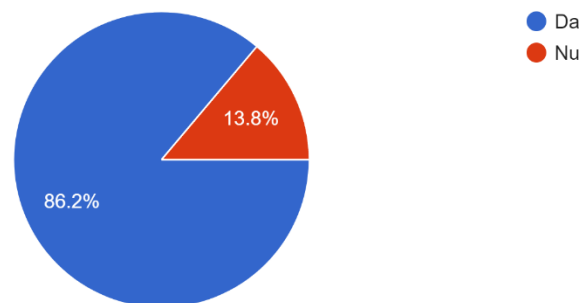


Figura 4

De altfel, cu ajutorul figurilor 3 și 4 observăm faptul că 86,2% dintre persoanele chestionate au simțit nevoia de ore suplimentare în perioada studiilor și doar 13,8% nu au simțit nevoia, persoanele întrebate având ca ultimă formă de învățământ finalizată școala primară, școala gimnazială, liceu și facultate. Astfel, putem spune că nevoia orelor suplimentare apare indiferent de forma de învățământ, indiferent dacă urmează un examen important precum bacalaureatul sau nu.

Este evidențiată nevoia unei schimbări în rândul elevilor și al studenților, pentru facilitarea procesului de învățare al acestora și ajutorarea profesorilor cărora le este greu ca într-un curs să ajungă la 100% numărul elevilor care au înțeles și nu mai au nevoie de pregătire și explicații în plus.

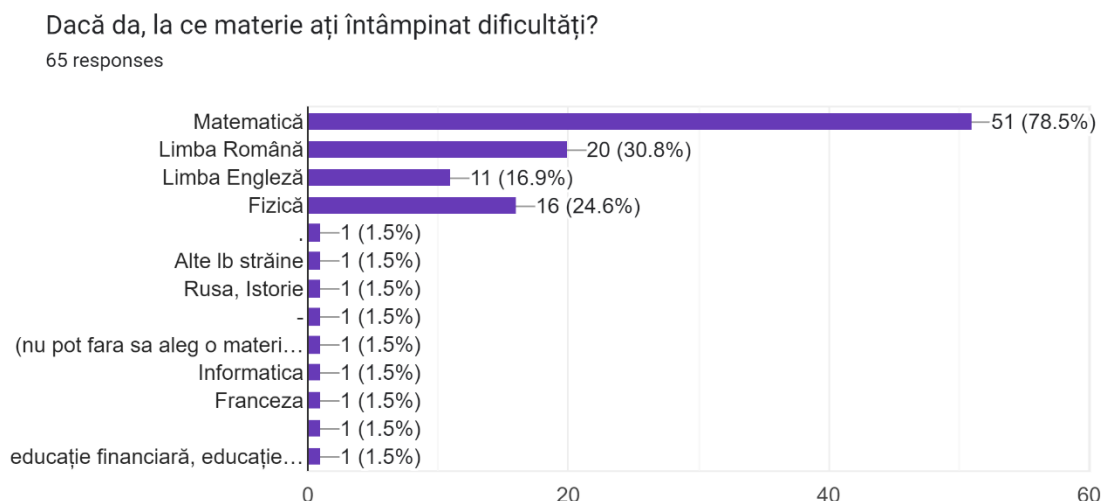


Figura 5

Ați apelat la meditații pe parcursul anilor de învățământ din cauza nevoii de suplimentare a orele de la clasă?

65 responses

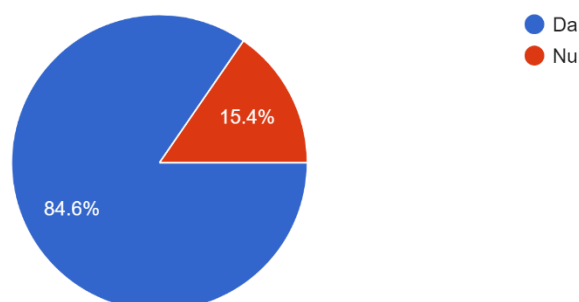


Figura 6

Din figura 5 reiese faptul ca doar 4 persoane din 65 nu au intampinat dificultati in perioada scolii. Consider ca numarul persoanelor ce nu au avut probleme pe parcursul anilor de studii este unul extraordinar de mic si trebuie schimbat. Acest fapt m-a motivat in realizarea aplicatiei de invatare suplimentara si sper sa reusesc sa aduc o imbunatatire si un beneficiu atat elevilor cat si profesorilor.

De asemenea, privind figura 6 observam ca 84,6% dintre persoanele chestionate au apelat la meditatii din cauza nevoi de a suplimenta orele de la clasa si doar 15,4% nu au facut meditatii. Procentul persoanelor care au facut meditatii este unul foarte mare, de altfel si nevoia si de pregatire suplimentara este uriasa. Sustin faptul ca platforma de invatare suplimentara poate sprijini nevoile elevilor.



Sunteți de acord că meditațiile NU sunt la îndemâna oricui?

65 responses

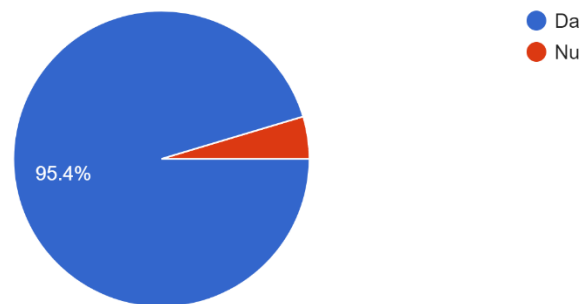


Figura 7

Figura 7 reprezintă un adevăr foarte trist cu care, din păcate, multe persoane se confruntă. Meditațiile din nefericire nu sunt la îndemână oricui, de altfel mulți elevi chiar dacă au nevoie de pregătire suplimentară nu pot avea parte de ea iar acest lucru îi poate împiedica în dezvoltarea lor.

Sunt de părere că toată lumea ar trebui să aibă acces la pregătire suplimentară în caz de nevoie și consider că platforma prezentată poate ajuta prin scăderea numărului elevilor care nu au acces la pregătire suplimentară și a celor care apelează la meditații.

Considerați că platformele de învățare online (ex Google Classroom, Moodle etc.) sunt benefice atât pentru profesori cât și pentru elevi/studenți?

65 responses

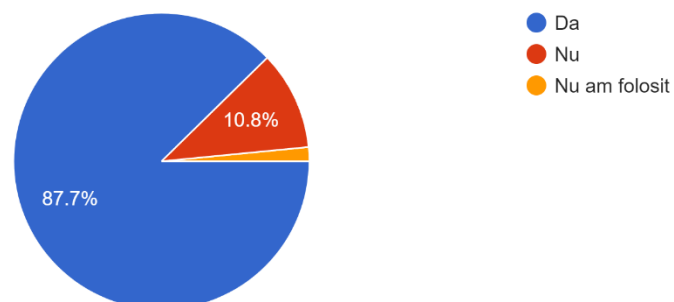


Figura 8

Considerați că o platformă online de învățare suplimentară, nouă și ușor de folosit, poate reduce numărul elevilor și al studenților care apelează la meditații ?

65 responses

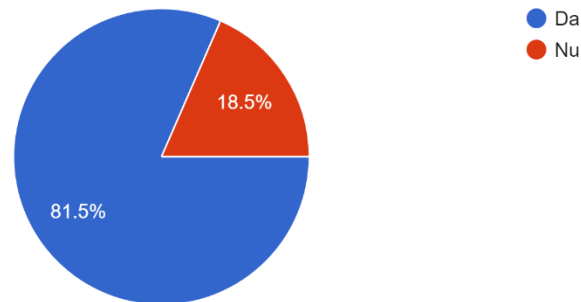


Figura 9

Analizand figurile 8 si 9 vedem parerea persoanelor chestionate referitor la platformele online si beneficiile lor. In figura 8 se observa ca doar 10,8% nu considera benefice aplicatiile de invatare online deja existente pe piata si 1,5% nu au folosit, ramanand un procentaj foarte mare de persoane ce considera benefice aplicatiile de educatie online.

De asemenea, in figura 9 este precizat faptul ca 81,5% dintre persoane sunt de parere ca o noua platforma online de invatare suplimentara poate reduce numarul elevilor care apeleaza la meditatii intarind ideea pe care o sustin si obiectivul pe care mi l-am propus in realizarea aplicatiei prezentate.

## 4 SOLUȚIA PROPUȘĂ

Abordarea pe care am ales să merg provine din documentarea pieței la momentul actual. Astfel după o analiză amănunțită am observat că o bună parte din marile companii precum JPMorgan Chase Bank, Intellibus, Deutsche Bank și multe altele folosesc Java Spring Boot pentru backend. Pentru frontend am ales să merg pe Angular folosind typescript cu toate că React este mai popular în rândul aplicațiilor mici dar pentru proiectele mari este mai folosit Angular. Pentru baza de date am ales să merg pe o bază de date relațională, respectiv MySQL, aceasta fiind open-source, de asemenea fiind cunoscută pentru performanță și scalabilitate.

### 4.1 Spring Boot

Spring este cel mai popular framework de java. Titlatura de cel mai popular framework este obținută din viteza, simplitatea și productivitatea acestuia.

Spring este un framework pentru a construi aplicații java, open-source, care oferă o infrastructură complexă și ușor de menținut pentru dezvoltarea aplicațiilor. Principalele avantaje în folosirea Spring:

1. Inversion of control – acesta este un principiu de proiectare care oferă modularitate programului, oferind frameworkului control asupra unor porțiuni de cod sau a unor obiecte. Acest lucru este realizabil prin Dependency Injection. Dependency Injection este mecanismul prin care controlul este inversat de la programator la framework.
2. Accesul la date – simplificarea accesului la date folosind tehnologii precum Hibernate, JPA (Java persistence API), MyBatis, EclipseLink. Acestea reduc complexitatea interacțiunii cu baza de date.
3. Ideal pentru construirea de RESTful API datorită anotărilor simpliste: @RestController, @GetMapping, @PostMapping simplifică definiția și gestionarea. Aceste anotări fac codul ușor de citit și întreținut permițând o dezvoltare ușor facilă a endpointurilor de tip CRUD (create, read, update, delete).
4. Securitatea – securitatea este ușor de facilitat datorită frameworkului Spring Security JWT (json-web token) care ajută în autentificare și autorizare. De asemenea oferă protecție împotriva atacurilor de tip CSRF

#### 4.1.1 Acces la date

În soluția mea am ales să folosesc JPA împreună cu hibernate. JPA fiind o soluție de a controla accesul la baza de date iar hibernate implementarea acestuia. Aceasta abstractizează toată complexitatea implementării în spatele unor interfețe și reguli.

JPA și hibernate sunt un mod de programare object-relational mapping, acesta fiind un mod de a alinia codul cu baza de date. Se structurează foarte bine datorită modului de programare orientat pe obiecte, fiecare clasă din codul java este mapată la un tabel.

Am ales sa folosesc acest model la acces de date fata de un model ca MyBatis pentru optimizarea si abstractizarea pe care o aduce jpa si hibernate acestea avand incorporat caching.

Cu toate ca MyBatis ofera mai mult control asupra interogarilor sql, am decis ca automatizarea ORM este de ajutor pentru mine datorita faptului ca mentinerea interogarilor este mult mai usor de facilitat.

#### 4.1.2 Securitate

Pentru partea de securitate folosesc Spring Security si json-web token pentru autentificare si autorizarea diferitelor roluri din aplicatie. Odata ce userul se autentifica acestuia ii este generat un jwt token care este trimis din frontend inapoi, in fiecare cerere pe care acesta o face, pentru a confirma ca acesta este insusi userul pentru care tokenul a fost generat.

##### Explicație Detaliată

**Autentificare:** Utilizatorul introduce credențialele (de exemplu, nume de utilizator și parolă) și trimite o cerere de autentificare către server.

Serverul verifică credențialele utilizatorului. Dacă autentificarea este reușită, se generează un token JWT care poate contine informatii despre utilizator si rolurile acestuia. Aceste informatii se numesc claims.

Tokenul JWT este semnat folosind o cheie secretă pentru a asigura integritatea și autenticitatea acestuia.

Tokenul JWT generat este trimis către frontend și stocat de obicei in memoria locala a browserului sau un cookies.

##### Autorizare:

La fiecare cerere ulterioara din frontend este trimis alaturi de informatiile cererii in antet tokenul. Serverul verifica tokenul asupra validitatii, daca tokenul este valid acesta permite cererea altfel nu trimitand o eroare.

## 4.2 Angular

Angular este un framework open-source bazat pe TypeScript, care rulează pe Node.js și este utilizat pentru crearea aplicațiilor web.

### De ce TypeScript și nu JavaScript?

Unul dintre cele mai mari avantaje pe care consider ca le are typescript asupra javascript este pentru ca acesta are erori de compilare datorita tipizarii statice in loc de tipizarea dinamica in javascript.

### Caracteristici ale Angular

Angular este bazat pe componente. Componentele ofera utilizatorului abilitatea de a defini propriile componenete personalizate. Aceste pot fi folosite simultan cu elementele HTML standard, precum `<div></div>`. Un exemplu de folosire al acestei caracteristici este crearea unei componente `<add-problem></add-problem>` care este o pagina cu un formular pentru adaugare a exercitiilor. Aceste poate fi folosit ca un „obiect” in orice alta componenta a aplicatiei. Astfel acest stil de programare este similar cu programarea orientata pe obiecte, aducand toate beneficiile ale acestuia cum ar fi modularitatea, reutilizarea, intretinerea si posibilitatea de extindere.

Printre beneficiile majore ale utilizării componentelor se numără:

- Organizare și Claritate: Codul este mult mai bine organizat și clar, facilitând înțelegerea și navigarea.
- Reutilizarea Codului: Componentele pot fi reutilizate în diferite părți ale aplicației, reducând redundanța și economisind timp.
- Întreținere Facilă: Codul modular și componentizat este mai ușor de întreținut și actualizat, permițând implementarea mai rapidă a modificărilor și a noilor funcționalități.

### 4.3 MySQL

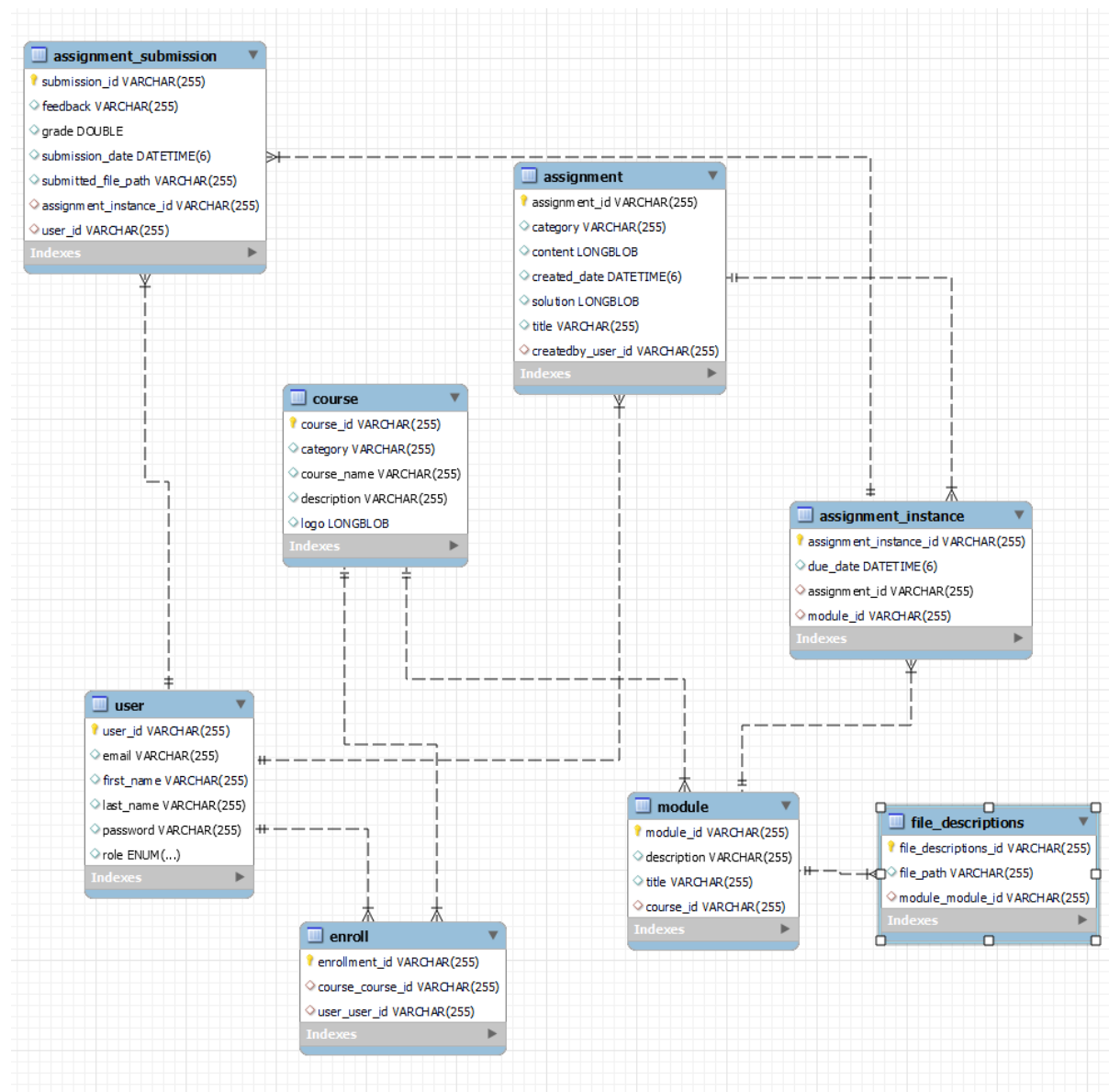
MySQL este un sistem de gestionare a bazelor de date relaționale (RDBMS) open-source. Este unul dintre cele mai populare și utilizate motoare de baze de date din lume, cunoscut pentru performanța sa, fiabilitatea și ușurința în utilizare. MySQL este dezvoltat, distribuit și susținut de Oracle.

### 4.4 Open-Source

Am menționat faptul ca alegerile mele in construirea aplicatiei sunt open-source. Astfel voi argumenta care sunt avantajele:

1. Transparența: dezvoltatorii pot inspecta codul sursa pentru a observa daca acesta este functional, de asemenea acesta este un factor bun si din punct de vedere al securitatii.
2. Securitatea: datorita transparenței dezvoltatorii pot analiza codul de vulnerabilitati ascunse. Legea lui Linus Torvalds „Given enough eyeballs, all bugs are shallow” – aceasta inseamna cu cat mai multa lume lucreaza si testeaza aplicatia problemele aplicatiei dispar.
3. Valoare educationala
4. Suport si comunitate: datorita faptului ca este open-source suportul si comunitatea este pe masura fiind foarte usor sa gasesti informatii. De asemenea toti dezvoltatorii pot aduce contributii.
5. Inovare si dezvoltare rapida: datorita faptului ca toata lumea poate contribuii duce la dezvoltarea rapida si aducerea a noi inovatii in tehnologie.

## 4.5 Arhitectura bazei de date



1. Tabela user: Aceasta contine informatiile despre utilizatori. Coloanele sunt: user\_id cheie primara, email first\_name last\_name, password si role.
2. Tabela course: Aceasta contine informatiile despre cursuri. Coloanele sunt course\_id cheie primara, category, course\_name, description si logo.
3. Tabela module: Aceasta contine informatii despre modulele din curs, fiecare modul fiind legat de un curs. Coloanele sunt module\_id cheie primara, description, title si course\_id cheie straina.
4. Tabela file\_description: Aceasta contine caile pentru fisierele din module. Coloanele sunt: file\_descriptions\_id cheie primara, file\_path, module\_module\_id cheie straina.
5. Tabela assignment: Aceasta contine informatii despre exercitiile care pot fi folosite ca teme. Coloane: assignment\_id cheie primara, category, content, created\_date, solution, title si createdby\_user\_id cheie straina

6. Tabela `assignment_instance`: Aceasta tabela contine informatii despre teme, fiind legata de module. Coloane: `assignment_instance_id` cheie primara, `due_date`, `assignment_id` cheie straina, `module_id` cheie straina
7. Tabela `assignment_submission`: Aceasta tabela contine informatii despre temele incarcate ale studentilor. Coloane: `submission_id` cheie primara, `feedback`, `grade`, `submission_date`, `submitted_file_path`, `assignment_instance_id` cheie straina si `user_id` cheie straina.
8. Tabela `enroll`: In aceasta tabela este retinut ce utilizator este inregistrat la un anumit curs. Coloane `enrollment_id` cheie primara, `course_course_id` cheie straina si `user_user_id` cheie straina.

## 4.6 Docker

Docker este o platformă open-source care automatizează deployment-ul aplicațiilor, permițând rularea acestora într-un mediu izolat. Docker facilitează împachetarea unei aplicații într-un singur container, ceea ce contribuie la principiile de dezvoltare continuă și livrare continuă.

În aplicație vor fi trei servicii: frontend, backend și baza de date, orchestrate printr-un fișier `docker-compose.yml`. Fiecare serviciu rulează într-un container propriu și este interconectat printr-o rețea comună. Astfel, vor rula trei containere:

1. `mysql`: Containerul care rulează MySQL.
2. `Spring Boot RESTful Web Services`: Containerul care rulează backend-ul aplicației, configurat pentru a se conecta la baza de date și la frontend.
3. `frontend`: Containerul care rulează serverul Angular pentru partea de frontend.

De asemenea, utilizez o rețea comună, `springboot-mysql-net`, pentru a permite containerelor să comunice eficient între ele.

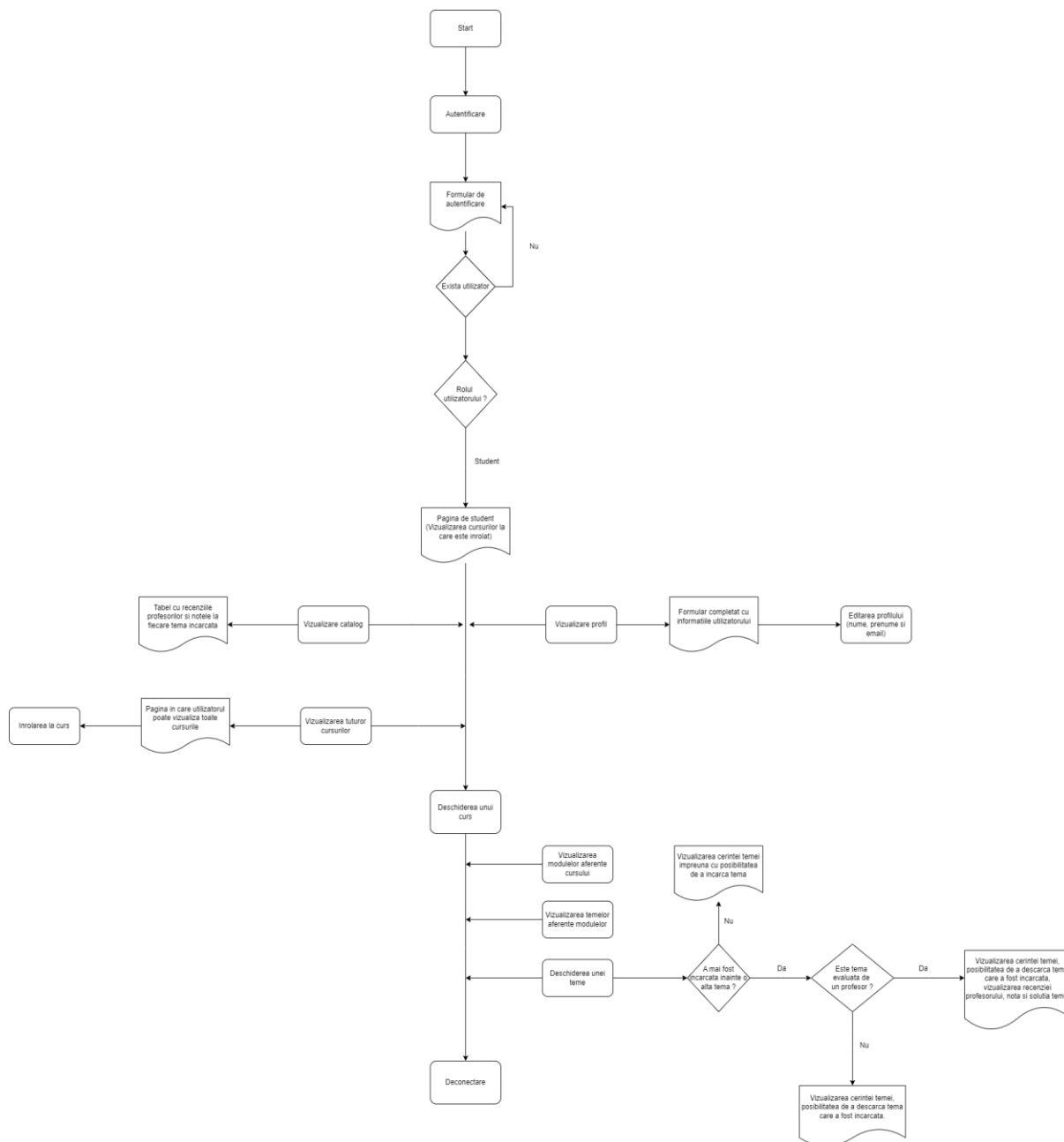
## 4.7 Principiul dezvoltare continua si livrare continua

Obiectivul principiului dezvoltare continua si livrarea continua este de a crește rapiditatea cu care este dezvoltată o aplicație și de a îmbunătăți calitatea produsului finit. Principalul atu al acestui principiu este faptul că îmbunătățește experiența dezvoltatorului dar și a consumatorului aplicației aplicația fiind livrată către acesta în cicluri stabilite de timp permitând o mai bună comunicare între aceștia. Consumatorul poate propune schimbări în aplicație la orice moment de timp, iar dezvoltatorul poate implementa această schimbare și o poate livra într-o nouă interacție.

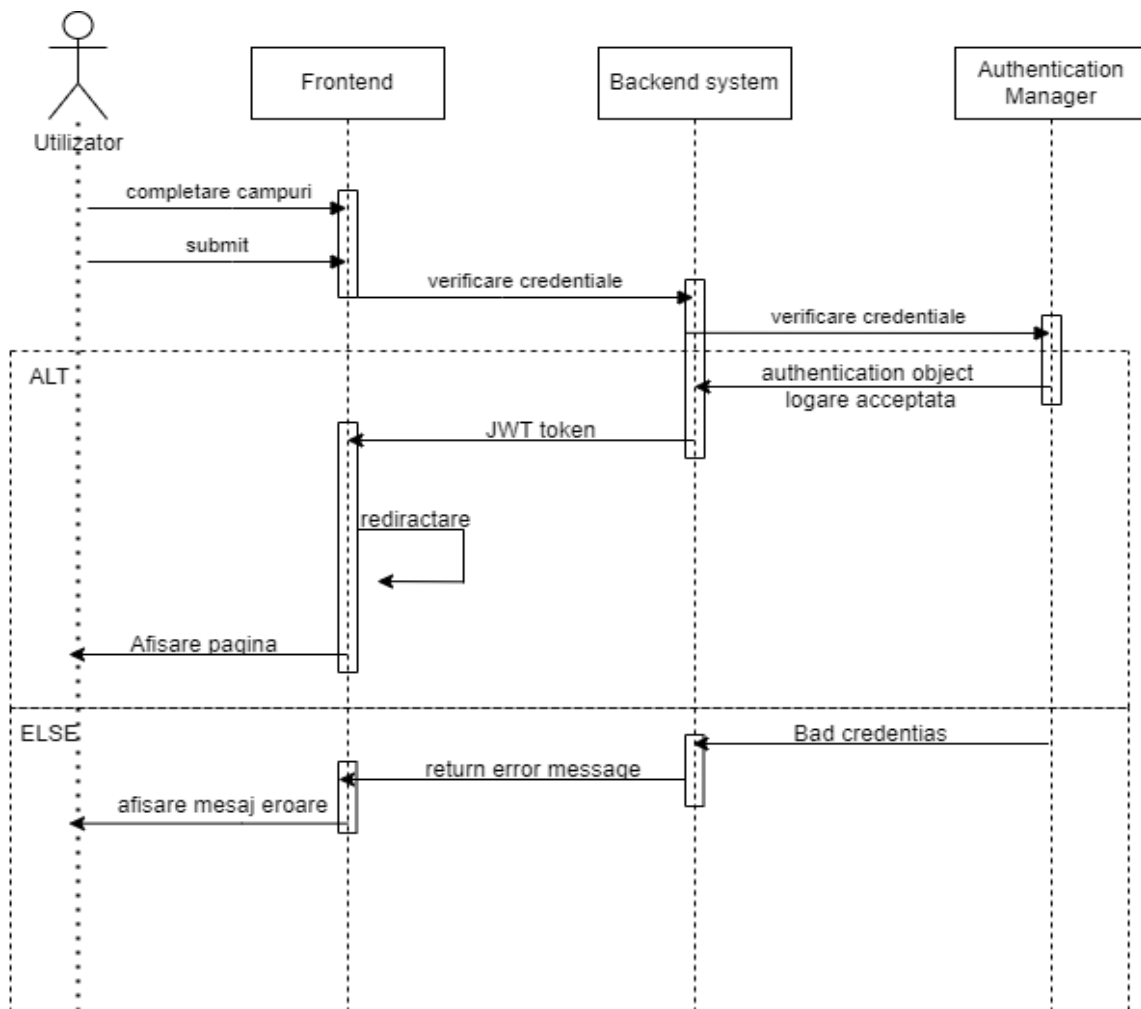
Astfel acest principiu se pleacă cel mai bine pe un mod de dezvoltare iterativ.

## 4.8 Descrierea fluxului functionalitatilor

Am ales să folosesc o diagrama simpla pentru a evidenta punctele critice de decizie si pentru a ilustra fluxul funcționalităților. Aceste diagrame sunt vitale pentru înțelegerea modului în care diferitele părți ale unui sistem lucrează împreună pentru a îndeplini anumite funcționalități. Diagramele de secvență oferă o imagine vizuală clară a ordinii mesajelor și a interacțiunilor dintre obiectele sistemului, făcând procesul mai ușor de urmărit și de găsit probleme sau îmbunătățiri.



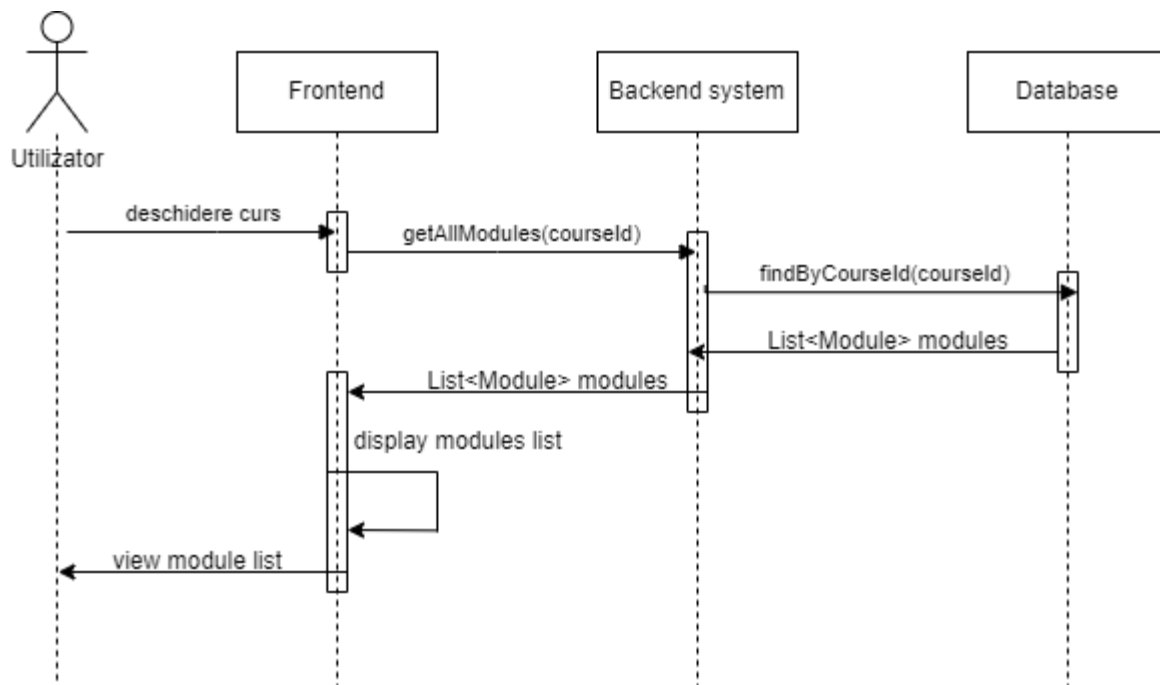




Interacțiunile dintre utilizator, front-end, back-end și managerul de autentificare sunt ilustrate în diagrama de secvență a fluxului de autentificare din figura X. Utilizatorul completează câmpurile (email și parolă) și apasă pe submit. Din front-end se trimite o cerere de tip POST către back-end. Back-end-ul folosește Authentication Manager pentru a verifica credențialele în baza de date.

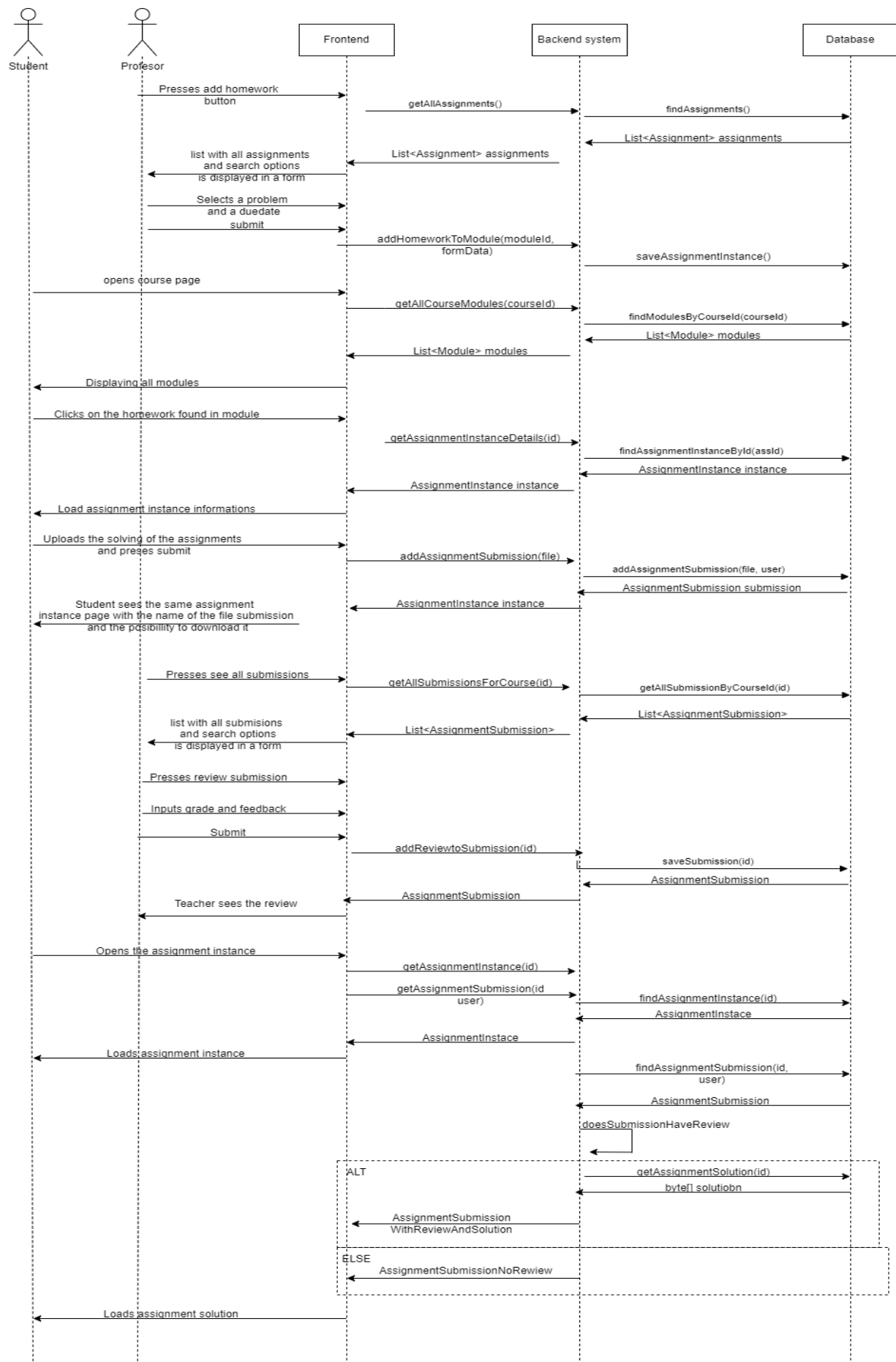
Dacă credențialele sunt corecte, se întoarce un obiect de autentificare. În back-end se generează un token JWT care conține proprietățile utilizatorului, cum ar fi rolul și emailul. Front-end-ul redirecționează utilizatorul pe pagina specifică rolului acestuia: pentru student și profesor pe pagina principală unde sunt afișate cursurile, iar pentru admin pe pagina specială pentru administrator și tokenul este salvat în local storage-ul browserului.

Dacă credențialele sunt incorecte, se întoarce un mesaj HTTP "Bad credentials", iar utilizatorului i se afișează un mesaj de eroare.



Interacțiunile dintre utilizator, front-end, back-end și baza de date sunt ilustrate în diagrama de secvență a fluxului de accesare a unui curs din figura X. Utilizatorul deschide cursul pentru a vizualiza lista de module disponibile. Din front-end se trimite o cerere de tip GET către back-end. Back-end-ul folosește baza de date pentru a găsi modulele asociate cursului specificat.

Dacă cererea este validă, back-end-ul interoghează baza de date pentru a găsi toate modulele asociate cursului cu ID-ul specificat. Baza de date returnează o listă de module către back-end. Back-end-ul trimite lista de module către front-end, care o afișează utilizatorului.

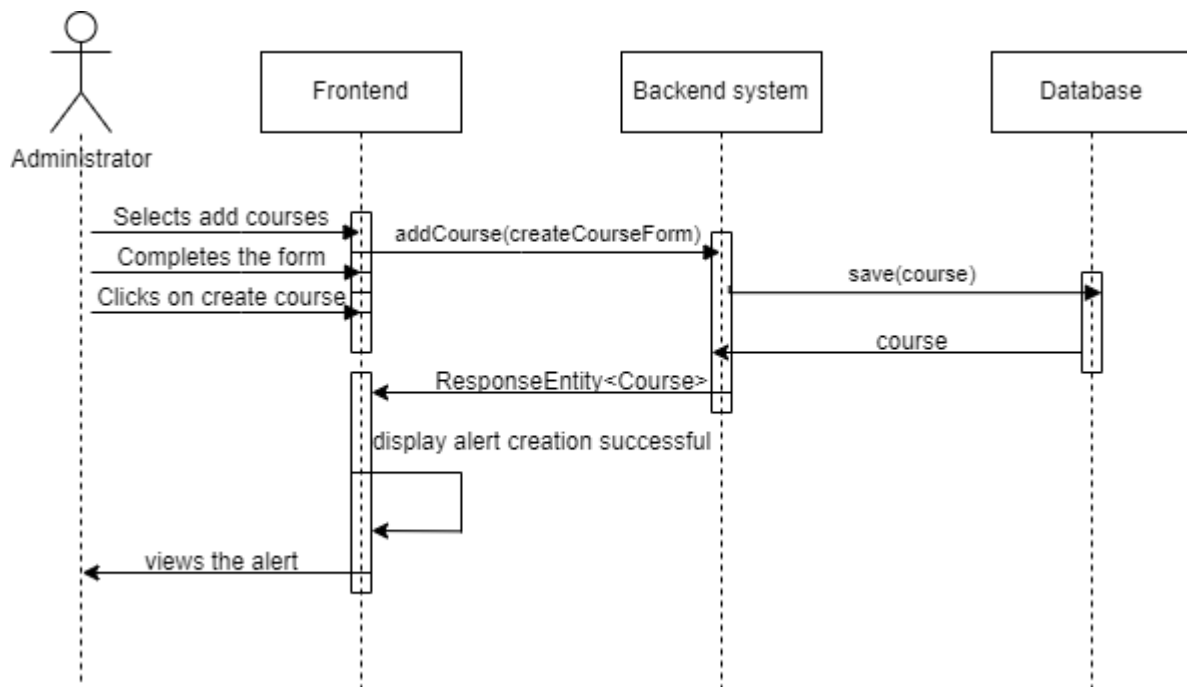


În diagrama din figura x este prezentată interacțiunea a diferitelor roluri, student și profesor, cu sistemul. Profesorul este în pagina de curs, acesta apasă pe butonul add homework aflat în unul dintre module. Îi este afișat un formular în care are un modul de cautare și toate problemele existente în lista de exerciții. Acesta alege unul dintre exerciții după care selectează un termen limită pentru încărcarea temei și apasă pe butonul add homework, pagina de curs este updatată, fiind vizibilă noua temă propusă.

Studentul deschide pagina de curs, vede tema propusă de profesor într-un modul acesta apasă click pe temă. O nouă pagină se deschide. Aceasta conține informațiile temei. Elevul termină tema și se decide să o încarce. Apasă pe butonul upload homework alege fișierul și da submit.

Profesorul apasă pe butonul „See all submissions”, astfel acesta este redirecționat pe o pagină care conține toate încărcările studenților. Apasă pe butonul de review, se deschide o pagină cu formular de unde acesta poate descărca ceea ce a încărcat studentul. Acesta completează recenzia și apasă pe butonul de submit.

Studentul deschide tema, acum este disponibilă recenzia profesorului, nota cât și rezolvarea corectă a temei. Dacă nu ar fi fost încă adăugată o recenzie acesta avea vizibil decât ultima lui încărcare pe care acesta putea să o descarce și să vadă ce se afla.



Doar utilizatorii cu rol de administrator pot crea cursuri. Aceștia când se autentifică sunt redirecționați automat către o pagină la care doar administratorii au acces. De altfel pentru adăugarea unui curs completează informațiile cursului cum ar fi denumirea, categoria și descrierea și apasă pe butonul de creează. Se face o cerere de tip POST către backend prin care se trimit informațiile completate în formular. Se salvează în baza de date, iar apoi se

introduce un obiect de tip ResponseEntity cu STATUS 201 si cursul care a fost adaugat. Apoi utilizatorul vede in aplicatie un mesaj „Cursul a fost adaugat cu succes”.

## **5 DETALII DE IMPLEMENTARE**

În plus fata de capitolul precedent acesta conține elemente specifice ale rezolvării problemei care au presupus dificultăți deosebite din punct de vedere tehnic. Pot fi incluse configurații, secvențe de cod, pseudo-cod, implementări ale unor algoritmi, analize ale unor date, scripturi de testare. De asemenea, poate fi detaliat modul în care au fost utilizate tehnologiile introduse în capitolul 3.

Criterii pentru calificativul Nesatisfăcător:

- Sunt prezentate pe scurt scheme și pseudo-cod.

Criterii pentru calificativul Satisfăcător:

- Descriere sumara a implementării, prezentarea unor secvențe nerelevante de cod, scheme, etc.

Criterii pentru calificativul Bine:

- Descrierea detaliată a algoritmilor/structurilor utilizați; Prezentarea etapizată a dezvoltării, inclusiv cu dificultăți de implementare întâmpinate, soluții descoperite; (dacă este cazul) demonstrarea corectitudinii algoritmilor utilizați.

### **5.1 Indicații formatare tabele**

Se recomandă utilizarea tabelelor de forma celui de mai jos. Font: Calibri, 9.

Orice tabel prezent în teză va fi referit în text; exemplu: a se vedea Tabel 1.

Tabel 1 - Sumarizare criterii

Calificativ	Criteriu	Observații
<b>Nesatisfăcător</b>	Sunt prezentate pe scurt scheme și pseudo-cod	
<b>Satisfăcător</b>	Descriere sumara a implementării, prezentarea unor secvențe nerelevante de cod, scheme, etc.	
<b>Bine</b>	Descrierea detaliată a algoritmilor/structurilor utilizați; Prezentarea etapizată a dezvoltării, inclusiv cu dificultăți de implementare întâmpinate, soluții descoperite; (dacă este cazul) demonstrarea corectitudinii algoritmilor utilizați.	Pot fi incluse configurații, secvențe de cod, pseudo-cod, implementări ale unor algoritmi, analize ale unor date, scripturi de testare.

## 6 STUDIU DE CAZ / EVALUAREA REZULTATELOR

Acest capitol trebuie să răspundă, în principiu, la **2 întrebări** și să se încheie cu **o discuție** a rezultatelor obținute. Cele doua întrebări la care trebuie să se răspundă sunt:

1) **Merge corect?** (Conform specificațiilor extrase în capitolul 2);

Evaluarea dacă merge corect se face pe baza cerințelor identificate în capitolele anterioare.

2) Cât de bine merge / cum se compară cu soluțiile existente? (pe baza unor metrici clare).

Evaluarea cât de bine merge trebuie să fie bazată pe procente, timpi, cantitate, numere, **comparativ cu soluțiile prezentate în capitolul 3**. Poate fi vorba de performanță, overhead, resurse consumate, scalabilitate etc.

În realizarea discuției, se vor utiliza tabele cu procente, rezultate numerice și grafice. În mod obișnuit, aici se fac comparații și teste comparative cu alte proiecte similare (dacă există) și se extrag puncte tari și puncte slabe. Se ține cont de avantajele menționate și se demonstrează viabilitatea abordării / aplicației, de dorit prin comparație cu alte abordări (dacă acest lucru este posibil). Cuvântul cheie la evaluare este „metrică”: trebuie să aveți noțiuni măsurabile și cuantificabile. În cadrul procesului de evaluare, explicați datele, tabelele și graficele pe care le prezentați și insistați pe relevanța lor, în următorul stil: „este de preferat ... deoarece ...”; explicați cititorului nu doar datele ci și semnificația lor și cum sunt acestea interpretate. Din această interpretare trebuie să rezulte poziționarea proiectului vostru printre alternativele existente, precum și cum poate fi acesta îmbunătățit în continuare.

Criterii pentru calificativul *Nesatisfăcător*:

- Aplicația este testată dar rulează pe calculatorul studentului, nu există posibilități de testare, nu a fost validată cu clienți / utilizatori;
- Nu au fost realizate comparații cu alte sisteme similare.

Criterii pentru calificativul *Satisfăcător*:

- [Dezvoltare de produs] Există teste unitare și de integrare, există o strategie de punere în funcțiune (*deployment*), există validare minimală cu clienții / utilizatorii.
- [Cercetare] Principalele componente și soluția în ansamblu au fost evaluate din punct de vedere al performanței, însă nu sunt folosite seturi de date standard, există unele erori de interpretare a datelor.
- [Ambele] Discuție minimală asupra relevanței rezultatelor prezentate, comparație minimală cu alte sisteme similare.

Criterii pentru calificativul *Bine*:

- [Dezvoltare de produs] Teste unitare și de integrare, instrumente de punere în funcțiune (*deployment*) utilizate și care arată lucru constant de-a lungul semestrului, lucrare validată cu clienții / utilizatorii, produs în producție.

- [Cercetare] Componentele și soluția în ansamblu au fost evaluate din punct de vedere al performanței, folosind seturi de date standard și cu o interpretare corectă a rezultatelor.
- [Ambele] Discuție cu prezentarea calitativă și cantitativă a rezultatelor, precum și a relevanței acestor rezultate printr-o comparație complexă cu alte sisteme similare.



## 7 CONCLUZII

În acest capitol este sumarizat întreg proiectul, de la obiective, la implementare, și la relevanța rezultatelor obținute. În finalul capitolului poate exista o subsecțiune de „Dezvoltări ulterioare”.

Criterii pentru calificativul *Nesatisfăcător*:

- Concluziile nu sunt corelate cu conținutul lucrării;

Criterii pentru calificativul *Satisfăcător*:

- Concluziile sunt corelate cu conținutul lucrării, însă nu se oferă o imagine asupra calității și relevanței rezultatelor obținute;

Criterii pentru calificativul *Bine*:

- Concluziile sunt corelate cu conținutul lucrării, și se oferă o imagine precisă asupra relevanței și calității rezultatelor obținute în cadrul proiectului.

## 8 BIBLIOGRAFIE

- Trebuie respectat **un singur standard** de trimiteri bibliografice (citare), **dintre** următoarele alternative:
  - APA (<http://pitt.libguides.com/c.php?g=12108&p=64730>)
  - IEEE (<https://iee-dataport.org/sites/default/files/analysis/27/IEEE%20Citation%20Guidelines.pdf>)
  - Harvard (<https://libweb.anglia.ac.uk/referencing/harvard.htm>)
  - Cu numerotarea referințelor în ordine alfabetică sau în ordinea apariției în text (de exemplu, stilul cu numere folosit de unele publicații ACM - <https://www.acm.org/publications/authors/reference-formatting>)
- Toate referințele din acest capitol trebuie să fie referite în text. Exemple:
  - [Articol jurnal]: [2];
  - [Articol conferință]: [3];
  - [Carte]: [4]
  - [Weblink]: [5]
  - [Application report] [6]

- [1] J. Silva-Martinez, "ELEN-325. Introduction to Electronic Circuits: A Design Approach," 2008. [Online]. Available: <http://www.ece.tamu.edu/~spalermo/ecen325/Section%20III.pdf>.
- [2] H. Baali, H. Djelouat, A. Amira and F. Bensaali, "Empowering Technology Enabled Care Using IoT and Smart Devices: A Review," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 1790-1809, 2018.
- [3] A. Haroon, S. Akram, M. A. Shah and A. Wahid, "E-Lithe: A Lightweight Secure DTLS for IoT," in *IEEE 86th Vehicular Technology Conference (VTC-Fall)*, Toronto, 2017.
- [4] A. K. Jain and R. C. Dubes., *Algorithms for Clustering Data*, Upper Saddle River: Prentice-Hall, Inc., 1988.
- [5] "Kernel panic! What are Meltdown and Spectre, the bugs affecting nearly every computer and device?," *techcrunch.com*, 2018. [Online]. Available: <https://techcrunch.com/2018/01/03/kernel-panic-what-are-meltdown-and-spectre-the-bugs-affecting-nearly-every-computer-and-device>. [Accessed 14 02 2018].
- [6] E. Rogers, "Understanding Buck-Boost Power Stages in Switch Mode Power Supplies," Texas Instruments, 2007.

- NU utilizați referințe la Wikipedia sau alte surse fără autor asumat.
- Pentru referințe la articole relevante accesibile în web (descrise prin URL) se va nota la bibliografie și data accesării.
- Mai multe detalii despre citarea referințelor din internet se pot regăsi la:
  - <http://www.writinghelp-central.com/apa-citation-internet.html>
  - <http://www.webliminal.com/search/search-web13.html>
- Note de subsol se utilizează dacă referiți un link mai puțin semnificativ o singură dată; Dacă nota este citată de mai multe ori, atunci utilizați o referință bibliografică.
- Dacă o imagine este introdusă în text și nu este realizată de către autorul lucrării, trebuie citată sursa ei (ca notă de subsol sau referință - este de preferat utilizarea unei note de subsol).
- Referințele se pun direct legate de text (de exemplu „KVM [1] uses”, „as stated by Popescu and Ionescu [12]”, etc.). Nu este recomandat să folosiți formulări de tipul „[1] uses”, „as stated in [12]”, „as described in [11]” etc.
- Afirmatiile de forma „are numerous”, „have grown exponentially”, „are among the most used”, „are an important topic” trebuie să fie acoperite cu citări, date concrete și analize comparative.
  - Mai ales în capitolele de introducere, „state of the art”, „related work” sau „background” trebuie să vă argumentați afirmațiile prin citări. Fiți autocritici și gândiți-vă dacă afirmațiile au nevoie de citări, chiar și cele pe care le considerați evidente.
  - Cea mai mare parte dintre citări vor fi în capitolele de introducere „state of the art”, „related work” sau „background”.
- Toate intrările bibliografice trebuie citate în text. Nu le adăugați pur și simplu la final.
- Nu copiați sau traduceți niciodată din surse de informație de orice tip (online, offline, cărți, etc.). Dacă totuși doriți să oferiți, prin excepție, un citat celebru - de maxim 1 frază- utilizați ghilimele și evident menționați sursa.
- Dacă reformulați idei sau creați un paragraf rezumat al unor idei folosind cuvintele voastre, precizați cu citare (referință bibliografică) sau cu notă de subsol sursa sau sursele de unde ați preluat ideile.

## 9 ANEXE

Anexele sunt opționale.

Ce poate intra în anexe:

- Exemplu de fișier de configurare sau compilare;
- Un tabel mai mare de ½ pagină;
- O figura mai mare de ½ pagină;
- O secvență de cod sursă mai mare de ½ pagină;
- Un set de capturi de ecran („screenshot”-uri);
- Un exemplu de rulare a unor comenzi plus rezultatul („output”-ul) acestora;
- În anexe intră lucruri care ocupă mai mult de o pagină ce ar întrerupe firul natural de parcurgere al textului.